WIRELESS VIDEO TRANSMISSION SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates generally to wireless transmission systems, and relates more particularly to a wireless video transmission system.

Developing an effective method for implementing enhanced television systems is a significant consideration for contemporary television designers and manufacturers. In conventional television systems, a display device may be utilized to view program information received from a program source. The conventional display device is typically positioned in a stationary location because of restrictions imposed by various physical connections that electrically couple the display device to input devices, output devices, and operating power. Other considerations such as display size and display weight may also significantly restrict viewer mobility in traditional television systems.

Portable television displays may advantageously provide viewers with additional flexibility when choosing an appropriate viewing location. For example, in a home environment, a portable television may readily be relocated to view programming at various remote locations throughout the home. A user may thus flexibly view television programming, even while performing other tasks in locations that are remote from a stationary display device.

However, portable television systems typically possess certain detrimental operational characteristics that diminish their effectiveness for use in modern television systems. For example, in order to eliminate restrictive physical connections, portable televisions typically receive television signals that are propagated from a remote terrestrial television transmitter to an antenna that is integral with the portable television. Because of the size and positioning constraints associated with a portable antenna, such portable televisions typically exhibit relatively poor reception characteristics, and the subsequent display of the transmitted television signals is therefore often of inadequate quality.

Other factors and considerations are also relevant to effectively implementing an enhanced wireless television system. For example, the evolution of digital data network technology and wireless digital transmission techniques may provide

additional flexibility and increased quality to portable television systems. However, current wireless data networks typically are not optimized for flexible transmission and reception of video information.

Furthermore, a significant proliferation in the number of potential program sources (both analog and digital) may benefit a system user by providing an abundance of program material for selective viewing. In particular, an economical wireless television system for flexible home use may enable television viewers to significantly improve their television-viewing experience by facilitating portability while simultaneously providing an increased number of program source selections.

However, because of the substantially increased system complexity, such an enhanced wireless television system may require additional resources for effectively managing the control and interaction of various system components and functionalities. Therefore, for all the foregoing reasons, developing an effective method for implementing enhanced television systems remains a significant consideration for designers and manufacturers of contemporary television systems.

A number of media playback systems use continuous media streams, such as video image streams, to output media content. However, some continuous media streams in their raw form often require high transmission rates, or bandwidth, for effective and/or timely transmission. In many cases, the cost and/or effort of providing the required transmission rate is prohibitive. This transmission rate problem is often solved by compression schemes that take advantage of the continuity in content to create highly packed data. Compression methods such Motion Picture Experts Group (MPEG) methods and its variants for video are well known. MPEG and similar variants use motion estimation of blocks of images between frames to perform this compression. With extremely high resolutions, such as the resolution of 1080i used in high definition television (HDTV), the data transmission rate of such a video image stream will be very high even after compression.

One problem posed by such a high data transmission rate is data storage. Recording or saving high resolution video image streams for any reasonable length of time requires considerably large amounts of storage that can be prohibitively expensive.

Another problem presented by a high data transmission rate is that many output devices are incapable of handling the transmission. For example, display systems that can be used to view video image streams having a lower resolution may not be capable of displaying such a high resolution. Yet another problem is the transmission of continuous media in networks with a limited bandwidth or capacity. For example, in a local area network with multiple receiving/output devices, such a network will often have a limited bandwidth or capacity, and hence be physically and/or logistically incapable of simultaneously supporting multiple receiving/output devices.

Laksono, U.S. Patent Application Publication Number 2002/0140851 A1 published October 3, 2002 discloses an adaptive bandwidth footprint matching for multiple compressed video streams in a limited bandwidth network.

Wang and Vincent in a paper entitled Bit Allocation and Constraints for Joint Coding of Multiple Video Programs, IEEE Transaction on Circuits and Systems for Video Technology, Vol. 9, No. 6, September 1999 discuss a multi-program transmission system in which several video programs are compressed, multiplexed, and transmitted over a single channel. The aggregate bit rate of the programs has to be equal to (or less than) the bandwidth (e.g., channel rate). This can be achieved by controlling either each individual program bit rate (independent coding) or the aggregate bit rate (joint coding). Thus in order to achieve such bit rate allocation, with a channel having 150 megabits/second of bandwidth, a first program may use 75 megabits/second, a second program may use 25 megabits/second, and a third program may use 50 megabits/second, with the channel bandwidth being distributed by measuring the bit-rate being transmitted.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates a gateway, media sources, receiving units, and a network.

FIG. 2 illustrates an analog extender.

FIG. 3 illustrate a digital extender.

FIG. 4 illustrates GOPs.

FIG. 5 illustrates virtual GOPs.

FIG. 6 illustrates a more detailed view of an extender.

FIG. 7 illustrates an analog source single stream.

FIG. 8 illustrates a digital source single stream.

FIG. 9 illustrates multiple streams.

FIG. 10 illustrates MPEG-2 TM5.

FIG. 11 illustrates dynamic rate adaptation with virtual GOPs.

FIG. 12 illustrates slowly varying channel conditions of super GOPS by GOP bit allocation.

FIG. 13 illustrates dynamic channel conditions of virtual super GOP by virtual super GOP bit allocation.

FIG. 14 illustrates dynamic channel conditions of super frame by super frame bit allocation.

FIG. 15 illustrates user preferences and priority determination for streams.

FIG. 16 illustrates the weight of a stream resulting from preferences at a particular point in time.

FIG. 17 illustrates the relative weight of streams set or changed at arbitrary times or on user demand.

FIG. 18 illustrates an MAC layer model.

FIG. 19 illustrates an APPLICATION layer model-based approach.

FIG. 20 illustrates an APPLICATION layer packet burst approach.

FIG. 21 illustrates ideal transmission and receiving.

FIG. 22 illustrate retransmission and fallback to lower data rates.

FIG. 23 illustrates pack submissions and packet arrivals.

FIG. 24 illustrates pack burst submissions and arrivals.

FIG. 25 illustrates packet burst submissions and arrivals with errors.

FIG. 26 illustrates measured maximum bandwidth using packet burst under ideal conditions.

FIG. 27 illustrates measured maximum bandwidth using packet burst under non-ideal conditions.

FIGS. 28A-28B illustrates receiving packets.

FIG. 29 illustrates transmitting packets.

BRIEF DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 illustrate a system for transmission of multiple data streams in a network that may have limited bandwidth. The system includes a central gateway media server 210 and a plurality of client receiver units 230, 240, 250. The central gateway media server may be any device that can transmit multiple data streams. The input data streams may be stored on the media server or arrive from an external source, such as a satellite television transmission 260, a digital video disc player, a video cassette recorder, or a cable head end 265, and are transmitted to the client receiver units 230, 240, 250 in a compressed format. The data streams can include display data, graphics data, digital data, analog data, multimedia data, audio data and the like. An adaptive bandwidth system on the gateway media server 210 determines the network bandwidth characteristics and adjusts the bandwidth for the output data streams in accordance with the bandwidth characteristics.

In one existing system, the start time of each unit of media for each stream is matched against the estimated transmission time for that unit. When any one actual transmission time exceeds its estimated transmission time by a predetermined threshold, the network is deemed to be close to saturation, or already saturated, and the system may select at least one stream as a target for lowering total bandwidth usage. Once the target stream associated with a client receiver unit is chosen, the target stream is modified to transmit less data, which may result in a lower data transmission rate. For example, a decrease in the data to be transmitted can be accomplished by a gradual escalation of the degree of data compression performed on the target stream, thereby reducing the resolution of the target stream. If escalation of the degree of data compression alone does not adequately reduce the data to be transmitted to prevent bandwidth saturation, the resolution of the target stream can also be reduced. For example, if the target stream is a video stream, the frame size could be scaled down, reducing the amount of data per frame, and thereby reducing the data transmission rate.

By way of background the bandwidth requirements for acceptable quality of different types of content vary significantly:

CD audio is generally transmitted at about 1 Mbps;

Standard definition video (MPEG-2) is generally transmitted at about 6 Mbps; High Definition video (MPEG-2) is generally transmitted at about 20 Mbps; and Multiple audio/video streams are generally transmitted at about 50-150 Mbps or more.

The overall quality can be expressed in many different ways, such as for example, the peak signal-to-noise ratio, delay (<100 ms for effective real-time two-way communication), synchronization between audio and video (<10 ms typically), and jitter (time varying delay). In many cases the audio/video streams are unidirectional, but may include a back-channel for communication.

There are many characteristics that the present inventors identified that may be considered for an audio/visual transmission system in order to achieve improved results over the technique described above.

(1) The devices may be located at different physical locations, and, over time, the users may change the location of these devices relative to the gateway. For example, the user may relocate the device near the gateway or farther away from the gateway, or, the physical environment may change significantly over time, both of which affect the performance of the wireless network for that device, and in turn the available bandwidth for other devices. This results in unpredictable and dynamically varying bandwidth.

(2) Different devices interconnected to the network have different resources and different usage paradigms. For example, different devices may have different microprocessors, different memory requirements, different display characteristics, different connection bandwidth capabilities, and different battery resources. In addition, different usage paradigms may include for example, a mobile handheld device versus a television versus a personal video recorder. This results in unpredictable and dynamically varying network maximum throughput.

(3) Multiple users may desire to access the data from the system simultaneously using different types of devices. As the user access data and stops accessing data the network conditions will tend to dynamically change. This results in unpredictable and dynamically varying network maximum throughput.

(4) Depending on the client device the transmitted data may need to be in different formats, such as for example, MPEG-2, MPEG-1, H.263, H.261, H.264, MPEG-4, analog, and digital. These different formats may have different impacts on the bandwidth. This results in unpredictable and dynamically varying network maximum throughput.

(5) The data provided to the gateway may be in the form of compressed bit streams which may include a constant bit rate (CBR) or variable bit rate (VBR). This results in unpredictable and dynamically varying network maximum throughput.

Various network technologies may be used for the gateway reception and transmission, such as for example, IEEE 802.11, Ethernet, and power-line networks (*e.g.*, HomePlug Powerline Alliance). While such networks are suitable for data transmission, they do not tend to be especially suitable for audio/video content because of the stringent requirements imposed by the nature of audio/video data transmission. Moreover, the network capabilities, and in particular the data maximum throughput offered, are inherently unpredictable and may dynamically change due to varying conditions described above. The data throughput may be defined in terms of the amount of actual (application) payload bits (per second) being transmitted from the sender to the receiver successfully. It is noted that while the system may refer to audio/video, the concepts are likewise used for video alone and/or audio alone.

With reference to one particular type of wireless network, namely, IEEE 802.11, such as IEEE 802.11a and 802.11b, they can operate at several different data link rates:

6, 9, 12, 18, 24, 36, 48, or 54 Mbps for 802.11(a), and

1, 2, 5.5, or 11 Mbps for 802.11(b).

However, the actual maximum throughput as seen by the application layer is lower due to protocol overhead and depends on the distance between the client device and the access point (AP), and the orientation of the client device. Accordingly, the potential maximum throughput for a device within a cell (*e.g.*, a generally circular area centered around the AP) is highest when the device is placed close to the AP and lower when it is farther away. In addition to the distance, other factors contribute to lowering the actual data maximum

throughput, such as the presence of walls and other building structures, and radio-frequency interference due to the use of cordless phones and microwave ovens. Furthermore, multiple devices within the same cell communicating with the same AP must share the available cell maximum throughput.

A case study by Chen and Gilbert, "Measured Performance of 5-GHz 802.11a wireless LAN systems", Atheros Communications, 27 August 2001 shows that the actual maximum throughput of an IEE 802.11a system in an office environment is only about 23 Mbps at 24 feet, and falls below 20 Mbps (approximately the rate of a single high definition video signal) at ranges over 70 feet. The maximum throughput of an 802.11(b) system is barely 6 Mbps and falls below 6 Mbps (approximately the rate of a single standard definition video signal at DVD quality) at ranges over 25 feet. The report quotes average throughput values for within a circular cell with radius of 65 feet (typical for large size houses in the US) as 22.6 Mbps and 5.1 Mbps for 802.11a and 802.11b, respectively. Accordingly, it may be observed that it is not feasible to stream a standard definition and a high definition video signal to two client devices at the same time using an 802.11a system, unless the video rates are significantly reduced. Also other situations likewise involve competing traffic from several different audiovisual signals. Moreover, wireless communications suffer from radio frequency interference from devices that are unaware of the network, such as cordless phones and microwave ovens, as previously described. Such interference leads to unpredictable and dynamic variations in network performance, *i.e.*, losses in data maximum throughput/bandwidth.

Wireless Local Area Networks (WLANs), such as 802.11 systems, include efficient error detection and correction techniques at the Physical (PHY) and Medium Access Control (MAC) layer. This includes the transmission of acknowledgment frames (packets) and retransmission of frames that are believed to be lost. Such retransmission of frames by the source effectively reduces the inherent error rate of the medium, at the cost of lowering the effective maximum throughput. Also, high error rates may cause the sending stations in the network to switch to lower raw data link rates, again reducing the error rate while decreasing the data rates available to applications.

Networks based on power-line communication address similar challenges due to the unpredictable and harsh nature of the underlying channel medium. Systems based on the HomePlug standard include technology for adapting the data link rate to the channel conditions. Similar to 802.11 wireless networks, HomePlug technology contains techniques such as error detection, error correction, and retransmission of frames to reduce the channel error rate, while lowering effective maximum throughput. Due to the dynamic nature of these conditions, the maximum throughput offered by the network may (*e.g.*, temporarily) drop below the data rate required for transmission of AV data streams. This results in loss of AV data, which leads to an unacceptable decrease in the perceived AV quality.

To reduce such limitations one may (1) improve network technology to make networks more suitable to audio/visual data and/or (2) one may modify the audio/visual data to make the audio/visual data more suitable to such transmission networks. Therefore, a system may robustly stream audio/visual data over (wireless) networks by:

(1) optimizing the quality of the AV data continuously, in real-time; and

(2) adapting to the unpredictable and dynamically changing conditions of the network.

Accordingly a system that includes dynamic rate adaption is suitable to accommodate distribution of high quality audio/video streams over networks that suffer from significant dynamic variations in performance. These variations may be caused by varying of distance of the receiving device from the transmitter, from interference, or other factors.

The following discussion includes single-stream dynamic rate adaptation, followed by multi-stream dynamic rate adaptation, and then various other embodiments.

Single Stream Dynamic Rate Adaptation

A system that uses dynamic rate adaptation for robust streaming of video over networks may be referred to as an extender. A basic form of an extender that processes a single video stream is shown in FIGS. 2 and 3. FIGS. 2 and 3 depict the

transmitting portion of the system, the first having analog video inputs, the second having digital (compressed) video inputs. The extender includes a video encoding or transcoding module, depending on whether the input video is in analog or digital (compressed) format. If the input is analog, the processing steps may include A/D conversion, as well as digital compression, such as by an MPEG-2 encoder, and eventual transmission over the network. If the input is already in digital format, such as an MPEG-2 bit stream, the processing may include transcoding of the incoming bit stream to compress the incoming video into an output stream at a different bit rate, as opposed to a regular encoder. A transcoding module normally reduces the bit rate of a digitally compressed input video stream, such as an MPEG-2 bit stream or any other suitable format.

The coding/transcoding module is provided with a desired output bit rate (or other similar information) and uses a rate control mechanism to achieve this bit rate. The value of the desired output bit rate is part of information about the transmission channel provided to the extender by a network monitor module. The network monitor monitors the network and estimates the bandwidth available to the video stream in real time. The information from the network monitor is used to ensure that the video stream sent from the extender to a receiver has a bit rate that is matched in some fashion to the available bandwidth (e.g., channel rate). With a fixed video bit rate normally the quality varies on a frame by frame basis. To achieve the optimal output bit rate, the coder/transcoder may increase the level of compression applied to the video data, thereby decreasing visual quality slowly. In the case of a transcoder, this may be referred to as transrating. Note that the resulting decrease in visual quality by modifying the bit stream is minimal in comparison to the loss in visual quality that would be incurred if a video stream is transmitted at bit rates that can not be supported by the network. The loss of video data incurred by a bit rate that can not be supported by the network may lead to severe errors in video frames, such as dropped frames, followed by error propagation (due to the nature of video coding algorithms such as MPEG). The feedback obtained from the network monitor ensures that the output bit rate is toward an optimum level so that any loss in quality incurred by transrating is minimal.

The receiver portion of the system may include a regular video decoder module, such as an MPEG-2 decoder. This decoder may be integrated with the network interface (*e.g.*, built into the hardware of a network interface card). Alternatively, the receiving device may rely on a software decoder (*e.g.*, if it is a PC). The receiver portion of the system may also include a counterpart to the network monitoring module at the transmitter. In that case, the network monitoring modules at the transmitter and receiver cooperate to provide the desired estimate of the network resources to the extender system. In some cases the network monitor may be only at the receiver.

If the system, including for example the extender, has information about the resources available to the client device consuming the video signal as previously described, the extender may further increase or decrease the output video quality in accordance with the device resources by adjusting bandwidth usage accordingly. For example, consider an MPEG-1 source stream at 4 Mbps with 640 by 480 spatial resolution at 30 fps. If it is being transmitted to a resource-limited device, *e.g.*, a handheld with playback capability of 320 by 240 picture resolution at 15 fps, the transcoder may reduce the rate to 0.5 Mbps by simply subsampling the video without increasing the quantization levels. Otherwise, without subsampling, the transcoder may have to increase the level of quantization. In addition, the information about the device resources also helps prevent wasting shared network resources. A transcoder may also convert the compression format of an incoming digital video stream, *e.g.*, from MPEG-2 format to MPEG-4 format. Therefore, a transcoder may for example: change bit rate, change frame rate, change spatial resolution, and change the compression format.

The extender may also process the video using various error control techniques, *e.g.* such methods as forward error correction and interleaving.

Dynamic Rate Adaptation

Another technique that may be used to manage available bandwidth is dynamic rate adaptation, which generally uses feedback to control the bit rate. The rate of the output video is modified to be smaller than the currently available network bandwidth

from the sender to the receiver, most preferably smaller at all times. In this manner the system can adapt to a network that does not have a constant bit rate, which is especially suitable for wireless networks.

One technique for rate control of MPEG video streams is that of the so-called MPEG-2 Test Model 5 (TM5), which is a reference MPEG-2 codec algorithm published by the MPEG group (see FIG. 10). Referring to FIG. 4, rate control in TM5 starts at the level of a Group-of-Pictures (GOP), consisting of a number of I, P, and B-type video frames. The length of a GOP in number of pictures is denoted by N_{GOP}. Rate control for a constant-bit-rate (CBR) channel starts by allocating a fixed number of bits G_{GOP} to a GOP that is in direct proportion to the (constant) bandwidth offered. Subsequently, a target number of bits is allocated to a specific frame in the GOP. Each subsequent frame in a GOP is allocated bits just before it is coded. After coding all frames in a GOP, the next GOP is allocated bits. This is illustrated in FIG. 4 where $N_{GOP} = 9$ for illustration purposes.

An extension for a time-varying channel can be applied if one can assume that the available bandwidth varies only slowly relative to the duration of a GOP. This may be the case when the actual channel conditions for some reason change only slowly or relatively infrequently. Alternatively, one may only be able to measure the changing channel conditions with coarse time granularity. In either case, the bandwidth can be modeled as a piece-wise constant signal, where changes are allowed only on the boundaries of a (super) GOP. Thus, G_{GOP} is allowed to vary on a GOP-by-GOP basis.

However, this does not resolve the issues when the bandwidth varies quickly relative to the duration of a GOP, *i.e.*, the case where adjustments to the target bit rate and bit allocation should be made on a frame-by-frame basis or otherwise a much more frequent basis. To allow adjustments to the target bit rate on a frame-by-frame basis, one may introduce the concept of a virtual GOP, as shown in FIG. 5 (see FIG. 11).

Each virtual GOP may be the same length as an actual MPEG GOP, any other length, or may have a length that is an integer multiple of the length of an actual MPEG GOP. A virtual GOP typically contains the same number of I-, P- and B-type pictures within a single picture sequence. However, virtual GOPs may overlap each other, where the next virtual GOP is shifted by one (or more) frame with respect to the current

virtual GOP. The order of I-, P- and B-type pictures changes from one virtual GOP to the next, but this does not influence the overall bit allocation to each virtual GOP. Therefore, a similar method, as used *e.g.* in TM5, can be used to allocate bits to a virtual GOP (instead of a regular GOP), but the GOP-level bit allocation is in a sense "re-started" at every frame (or otherwise "re-started" at different intervals).

Let R_t denote the remaining number of bits available to code the remaining frames of a GOP, at frame t. Let S_t denote the number of bits actually spent to code the frame at time t. Let N_t denote the number of frames left to code in the current GOP, starting from frame t.

In TM5, R_t is set to 0 at the start of the sequence, and is incremented by G_{GOP} at the start of every GOP. Also, S_t is subtracted from R_t at the end of coding a picture. It can be shown that R_t can be written as follows, in closed form:

$$R_t = N_t G_P + \sum_{j=1}^{t-1} \left(G_P - S_j \right) \quad ,(1)$$

where G_P is a constant given by:

$$G_P = \frac{G_{GOP}}{N_{GOP}}$$

indicating the average number of bits available to code a single frame.

To handle a time varying bandwidth, the constant G_P may be replaced by G_t, which may vary with t. Also, the system may re-compute (1) at every frame t, *i.e.*, for each virtual GOP. Since the remaining number of frames in a virtual GOP is N_{GOP}, the system may replace N_t by N_{GOP}, resulting in:

$$R_t = N_{GOP} G_t + \sum_{j=1}^{t-1} \left(G_j - S_j \right) \quad ,(2)$$

Given R_t, the next step is allocate bits to the current frame at time t, which may be of type I, P, or B. This step takes into account the complexity of coding a particular frame, denoted by C_t. Frames that are more complex to code, *e.g.*, due to complex object motion in the scene, require more bits to code, to achieve a certain quality. In TM-5, the encoder maintains estimates of the complexity of each type of frame (I, P, or B), which are updated after coding each frame. Let C_I, C_P, and C_B denote the current estimates of the complexity for I, P and B frames. Let N_I, N_P and N_B denote the number of frames of type I, P and B left to encode in a virtual GOP (note that these are constants in the case of virtual GOPs).

TM5 prescribes a method for computing T_I, T_P and T_B, which are the target number of bits for an I, B, or P picture to be encoded, based on the above parameters. The TM5 equations may be slightly modified to handle virtual GOPs as follows:

$$T_I = R_t \Big/ \left(N_I + N_P \frac{K_I C_P}{K_P C_I} + N_B \frac{K_I C_B}{K_B C_I} \right)$$

$$T_P = R_t \Big/ \left(N_P + N_I \frac{K_P C_I}{K_I C_P} + N_B \frac{K_P C_B}{K_B C_P} \right) \quad ,(3)$$

$$T_B = R_t \Big/ \left(N_B + N_I \frac{K_B C_I}{K_I C_B} + N_P \frac{K_B C_P}{K_P C_B} \right)$$

where K_I, K_P, and K_B are constants. I, B, P, refer to I frames, B frames, and P frames, and C is a complexity measure. It is to be understood that any type of compression rate distortion model, defined in the general sense, may likewise be used.

As it may be observed, this scheme permits the reallocation of bits on a virtual GOP basis from frame to frame (or other basis consistent with virtual GOP

spacing). The usage and bit allocation for one virtual GOP may be tracked and the unused bit allocation for a virtual GOP may be allocated for the next virtual GOP.

Multi-Stream Dynamic Rate Adaptation

The basic extender for a single AV stream described above will encode an analog input stream or adapt the bit rate of an input digital bit stream to the available bandwidth without being concerned about the cause of the bandwidth limitations, or about other, competing streams, if any. In the following, the system may include a different extender system that processes multiple video streams, where the extender system assumes the responsibility of controlling or adjusting the bit rate of multiple streams in the case of competing traffic.

The multi-stream extender, depicted in FIG. 6, employs a "(trans)coder manager" on top of multiple video encoders/transcoders. As shown in FIG. 6, the system operates on n video streams, where each source may be either analog (*e.g.* composite) or digital (*e.g.* MPEG-2 compressed bitstreams). Here, V_n denotes input stream n, while V'_n denotes output stream n. R_n denotes the bit rate of input stream n (this exists only if input stream n is in already compressed digital form; it is not used if the input is analog), while R'_n denotes the bit rate of output stream n.

Each input stream is encoded or transcoded separately, although their bit rates are controlled by the (trans)coder manager. The (trans)coder manager handles competing requests for bandwidth dynamically. The (trans)coding manager allocates bit rates to multiple video streams in such a way that the aggregate of the bit rates of the output video streams matches the desired aggregate channel bit rate. The desired aggregate bit rate, again, is obtained from a network monitor module, ensuring that the aggregate rate of multiple video streams does not exceed available bandwidth. Each coder/transcoder again uses some form of rate control to achieve the allocated bit rate for its stream.

In this case, the system may include multiple receivers (not shown in the diagram). Each receiver in this system has similar functionality as the receiver for the single-stream case.

As in the single-stream case, the bit rate of the multiple streams should be controlled by some form of bit allocation and rate control in order to satisfy such constraints. However, in the case of a multi-stream system, a more general and flexible framework is useful for dynamic bit rate adaptation. There are several reasons for this, as follows:

(1) The system should deal with multiple AV streams that may have different characteristics, and should allocate the available bits as supported by the channel accordingly;

(2) The system should deal with the network characteristics, which are partly unpredictable, and need special attention in the case of multiple receivers as described later;

(3) The system should handle any differences between the receiving devices themselves, such as differences in screen sizes, supported frame rates, etc.; and

(4) The different video sources may be regarded as different in importance due to their content. Also, since the different video streams are viewed by different people (users), possibly in different locations (*e.g.*, different rooms in a home), the system may want to take the preferences of the different users into account.

The resulting heterogeneity of the environment may be taken into account during optimization of the system.

To this end, the multi-stream extender system may optionally receive further information as input to the transcoder manager (in addition to information about the transmission channel), as shown in FIG. 6. This includes, for example:

Information about each receiving device;

Information about each video source; and

Information about the preferences of each user.

In the following subsections, first is listed the type of constraints that the bit rate of the multiple streams in this system are subject to. Then, the notion of stream prioritizing is described, which is used to incorporate certain heterogeneous characteristics of the network as discussed above. Then, various techniques are described to achieve multi-stream (or joint) dynamic rate adaptation.

Bit Rate Constraints For Multiple Streams

The bit rates of individual audio/video streams on the network are subject to various constraints.

Firstly, the aggregate rates of individual streams may be smaller than or equal to the overall channel capacity or network bandwidth from sender to receiver. This bandwidth may vary dynamically, due to increases or decreases in the number of streams, due to congestion in the network, due to interference, etc.

Further, the rate of each individual stream may be bound by both a minimum and a maximum. A maximum constraint may be imposed due to the following reasons.

(1) A stream may have a maximum rate due to limitations of the channel or network used. For instance, if a wireless network is used, the maximum throughput to a single device depends on the distance between the access point and the client device. Note that this maximum may be time-varying. For instance, if the client device in a wireless network is portable and its distance to the access point is increasing (*e.g.* while being carried), the maximum throughput is expected to decrease.

(2) A stream may have a maximum rate due to limitations of the client device. The client device may have

limited capabilities or resources, *e.g.*, a limited buffer size or limited processing power, resulting in an upper bound on the rate of an incoming audio/video stream.

(3) A stream may have a minimum rate imposed by the system or by the user(s), in order to guarantee a minimum quality. If this minimum rate cannot be provided by the system, transmission to the device may not be performed. This helps achieve some minimum quality. A stream may also have a minimum rate imposed in order to prevent buffer underflow.

Stream Prioritizing Or Weighting

The (trans)coder manager discussed above may employ several strategies. It may attempt to allocate an equal amount of available bits to each stream; however, in this case the quality of streams may vary strongly from one stream to the other, as well as in time. It may also attempt to allocate the available bits such that the quality of each stream is approximately equal; in this case, streams with highly active content will be allocated more bits than streams with less active content. Another approach is to allow users to assign different priorities to different streams, such that the quality of different streams is allowed to vary, based on the preferences of the user(s). This approach is generally equivalent to weighting the individual distortion of each stream when the (trans)coder manager minimizes the overall distortion.

The priority or weight of an audio/video stream may be obtained in a variety of manners, but is generally related to the preferences of the users of the client devices. Note that the weights (priorities) discussed here are different from the type of weights or coefficients seen often in literature that correspond to the encoding complexity of a macro block, video frame, group of frames, or video sequence (related to the amount of motion or texture variations in the video), which may be used to achieve a uniform quality among such parts of the video. Here, weights will purposely result in a non-uniform quality

distribution across several audio/video streams, where one (or more) such audio/video stream is considered more important than others. Various cases, for example, may include the following, and combinations of the following.

Case A

The weight of a stream may be the result of a preference that is related to the client device (see FIG. 15). That is, in the case of conflicting streams requesting bandwidth from the channel, one device is assigned a priority such that the distortion of streams received by this device are deemed more severe as an equal amount of distortion in a stream received by another device. For instance, the user(s) may decide to assign priority to one TV receiver over another due to their locations. The user(s) may assign a higher weight to the TV in the living room (since it is likely to be used by multiple viewers) compared to a TV in the bedroom or den. In that case, the content received on the TV in the living room will suffer from less distortion due to transcoding than the content received on other TVs. As another instance, priorities may be assigned to different TV receivers due to their relative screen sizes, *i.e.*, a larger reduction in rate (and higher distortion) may be acceptable if a TV set's screen size is sufficiently small. Other device resources may also be translated into weights or priorities.

Such weighting could by default be set to fixed values, or using a fixed pattern. Such weighting may require no input from the user, if desired.

Such weighting may be set once (during set up and installation). For instance, this setting could be entered by the user, once he/she decides which client devices are part of the network and where they are located. This set up procedure could be repeated periodically, when the user(s) connect new client devices to the network.

Such weighting may also be the result of interaction between the gateway and client device. For instance, the client device may announce and describe itself to the gateway as a certain type of device. This may result in the assignment by the gateway of a certain weighting or priority value to this device.

Case B

The weight of a stream may be result of a preference that is related to a content item (such as TV program) that is carried by a particular stream at a particular point in time (see FIG. 16). That is, for the duration that a certain type of content is transmitted over a stream, this stream is assigned a priority such that the distortion of this stream is deemed more severe as an equal amount of distortion in other streams with a different type of content, received by the same or other devices. For instance, the user(s) may decide to assign priority TV programs on the basis of its genre, or other content-related attributes. These attributes, *e.g.* genre information, about a program can be obtained from an electronic program guide. These content attributes may also be based on knowledge of the channel of the content (*e.g.* Movie Channel, Sports Channel, etc). The user(s) may for example assign a higher weight to movies, compared to other TV programs such as gameshows. In this case, when multiple streams contend for limited channel bandwidth, and one stream carries a movie to one TV receiver, while another stream simultaneously carries a gameshow to another TV, the first stream is assigned a priority such that it will be distorted less by transcoding than the second stream.

Such weighting could by default be set to fixed values, or using a fixed pattern. Such weighting may require no input from the user, if desired.

Such weighting may be set once (during set up and installation). For instance, this setting could be entered by the user, once he/she decides which type(s) of content are important to him/her. Then, during operation, the gateway may match the description of user preferences (one or more user preferences) to descriptions of the programs transmitted. The actual weight could be set as a result of this matching procedure. The procedure to set up user preferences could be repeated periodically. The user preference may be any type of preference, such as those of MPEG-7 or TV Anytime. The system may likewise include the user's presence (any user or a particular user) to select, at least in part, the target bit rate. The user may include direct input, such as a remote control. Also, the system may include priorities among the user preferences to select the target bit rate.

Such weighting may also be the result of the gateway tracking the actions of the user. For instance, the gateway may be able to track the type of content that the user(s)

consume frequently. The gateway may be able to infer user preferences from the actions of the user(s). This may result in the assignment by the gateway of a certain weighting or priority value to certain types of content.

Case C

The relative weight of streams may also be set or changed at arbitrary times or on user demand (see FIG. 17).

Such weighting may be bound to a particular person in the household. For instance, one person in a household may wish to receive the highest possible quality content, no matter what device he/she uses. In this case, the weighting can be changed according to which device that person is using at any particular moment.

Such weighting could be set or influenced at an arbitrary time, for instance, using a remote control device.

Such weighting could also be based on whether a user is recording content, as opposed to viewing. Weighting could be such that a stream is considered higher priority (hence should suffer less distortion) if that stream is being recorded (instead of viewed).

Case D

The relative weight of streams may also be set based on their modality. In particular, the audio and video streams of an audiovisual stream may be separated and treated differently during their transmission. For example, the audio part of an audiovisual stream may be assigned a higher priority than the video part. This case is motivated by the fact that when viewing a TV program, in many cases, loss of audio information is deemed more severe by users than loss of video information from the TV signal. This may be the case, for instance, when the viewer is watching a sports program, where a commentator provides crucial information. As another example, it may be that users do not wish to degrade the quality of audio streams containing hi-quality music. Also, the audio quality could vary among different speakers or be sent to different speakers.

Network Characteristics

The physical and data link layers of the aforementioned networks are designed to mitigate the adverse conditions of the channel medium. One of the characteristics of these networks specifically affects bit allocation among multiple streams as in a multi-stream extender system discussed here. In particular, in a network based on IEEE 802.11, a gateway system may be communicating at different data link rates with different client devices. WLANs based on IEEE 802.11 can operate at several data link rates, and may switch or select data link rates adaptively to reduce the effects of interference or distance between the access point and the client device. Greater distances and higher interference cause the stations in the network to switch to lower raw data link rates. This may be referred to as multi-rate support. The fact that the gateway may be communicating with different client devices at different data rates, in a single wireless channel, affects the model of the channel as used in bit allocation for joint coding of multiple streams.

Prior work in rate control and bit allocation uses a conventional channel model, where there is a single bandwidth that can simply be divided among AV streams in direct proportion to the requested rates for individual AV streams. The present inventors determined that this is not the case in LANs such as 802.11 WLANs due to their multi-rate capability. Such wireless system may be characterized in that the sum of the rate of each link is not necessarily the same as the total bandwidth available from the system, for allocation among the different links. In this manner, a 10 Mbps video signal, and a 20 Mbps video signal may not be capable of being transmitted by a system having a maximum bandwidth of 30 Mbps. The bandwidth used by a particular wireless link in an 802.11 wireless system is temporal in nature and is related to the maximum bandwidth of that particular wireless link. For example, if link 1 has a capacity of 36 Mbps and the data is transmitted at a rate of 18 Mbps the usage of that link is 50%. This results in using 50% of the systems overall bandwidth. For example, if link 2 has a capacity of 24 Mbps and the data is transmitted at a rate of 24 Mbps the usage of link 2 is 100%. Using link 2 results in using 100% of the system's overall bandwidth leaving no bandwidth for other links, thus only one stream can be transmitted.

Bit Allocation In Joint Coding Of Multiple Streams

A more optimal approach to rate adaptation of multiple streams is to apply *joint* bit allocation/rate control. This approach applies to the case where the input streams to the multi-stream extender system are analog, as well as the case where the input streams are already in compressed digital form.

Let the following parameters be defined:

N_L denote the number of streams

p_n denote a weight or priority assigned to stream n, with $p_n \geq 0$

a_n denote a minimum output rate for stream n, with $a_n \geq 0$

b_n denote a maximum output rate for stream n, with $b_n \geq a_n$

$D_n(r)$ denote the distortion of output stream n as a function of its output rate r (*i.e.* the distortion of the output with respect to the input of the encoder or transcoder)

R_C denote the available bandwidth of the channel or maximum network maximum throughput

R_n denotes the bit rate of input stream n

R'_n denotes the bit rate of output stream n

Note that R_n, R'_n and R_C may be time-varying in general; hence, these are functions of time t.

The problem of the multi-stream extender can be formulated generically as follows:

The goal is to find the set of output rates R'_n, n = 1, …, N_L, that maximizes the overall quality of all output streams or, equivalently, minimizes an overall distortion criterion D, while the aggregate rate of all streams is within the capacity of the channel.

One form of the overall distortion criterion D is a weighted average of the distortion of the individual streams:

$$D=\sum_{n=1}^{N_L} p_n D_n(R'_n) \tag{4}$$

Another form is the maximum of the weighted distortion of individual streams:

$$D = \max_n \{p_n D_n(R'_n)\} \qquad (5)$$

In this section, a conventional channel model is used, similar to cable tv, where an equal amount of bit rate offered to a stream corresponds to an equal amount of utilization of the channel, while it may be extended to the wireless type utilizations described above. Therefore, the goal is to minimize a criterion such as (4) or (5), subject to the following constraints:

$$\sum_{n=1}^{N_L} R'_n \le R_C \qquad (6)$$

and, for all n,

$$0 \le a_n \le R'_n \le b_n \le R_n \qquad (7)$$

at any given time t.

In the case of transcoding, note that the distortion of each output stream V'_n is measured with respect to the input stream V_n, which may already have significant distortion with respect to the original data due to the original compression. However, this distortion with respect to the original data is unknown. Therefore, the final distortion of the output stream with respect to the original (not input) stream is also unknown, but bounded below by the distortion already present in the corresponding input stream V_n. It is noted that in the case of transcoding, a trivial solution to this problem is found when the combined input rates do not exceed the available channel bandwidth, i.e, when:

$$\sum_{n=1}^{N_L} R_n \le R_C \qquad (8)$$

In this case, $R'_n = R_n$ and $D_n(R'_n) = D_n(R_n)$ for all n, and no transcoding needs to be applied.

It is noted that no solution to the problem exists, when:

$$R_C \leq \sum_{n=1}^{N_L} a_n \tag{9}$$

This may happen when the available channel bandwidth / network maximum throughput would drop (dramatically) due to congestion, interference, or other problems. In this situation, one of the constraints (7) would have to be relaxed, or the system would have to deny access to a stream requesting bandwidth.

It is noted that an optimal solution that minimizes the distortion criterion as in (5) is one where the (weighted) distortion values of individual streams are all equal.

It is noted that (6) embodies a constraint imposed by the channel under a conventional channel model. This constraint is determined by the characteristics of the specific network. A different type of constraint will be used as applied to LANs with multi-rate support.

A few existing optimization algorithms exist that can be used to find a solution to the above minimization problem, such as Lagrangian optimization and dynamic programming. Application of such optimization algorithms to the above problem may require search over a large solution space, as well as multiple iterations of compressing the video data. This may be prohibitively computationally expensive. A practical approach to the problem of bit allocation for joint coding of multiple video programs extends the approach used in the so-called MPEG-2 Test Model 5 (TM5).

An existing approach is based on the notions of *super GOP* and *super frame*. A normal MPEG-2 GOP (Group-of-Pictures) of a single stream contains a number of I, P and B-type frames. A super GOP is formed over multiple MPEG-2 streams and consists of N_{GOP} super frames, where a super frame is a set of frames containing one frame from each stream and all frames in a super frame coincide in time. A super GOP always contains an integer number of stream-level MPEG-2 GOPs, even when the GOPs of individual streams are not the same length and not aligned. The bit allocation method includes a target number of bits assigned to a super GOP. This target number T_s is the same for every super GOP and is derived from the channel bit rate, which is assumed fixed. Given T_s, the bit allocation is done for each super frame within a super GOP. The resulting

target number of bits for a super frame T_t depends on the number of I, P, and B frames in the given streams. Then, given T_t, the bit allocation is done for each frame within a super frame. The resulting target number of bits for a frame within a super frame at time t is denoted by $T_{t,n}$.

The existing technique is based on the use of a complexity measure C for a video frame, that represents the "complexity" of encoding that frame. Subsequently, streams are allocated bits proportional to the estimated complexity of the frames in each stream. That is, streams with frames that are more "complex" to code, receive more bits during bit allocation compared to streams that are less "complex" to code, resulting in an equal amount of distortion in each stream.

The complexity measure C for a video frame is defined as the product of the quantization value used to compress the DCT coefficients of that video frame, and the resulting number of bits generated to code that video frame (using that quantization value). Therefore, the target number of bits $T_{t,n}$ for a particular frame within a super frame can be computed on the basis of an estimate of the complexity of that frame, $C_{t,n}$, and the quantizer used for that frame, $Q_{t,n}$:

$$T_{t,n} = \frac{C_{t,n}}{Q_{t,n}} \tag{10}$$

The value of $C_{t,n}$ is assumed constant within a stream for all future frames of the same type (I, P or B) to be encoded. Therefore, $C_{t,n}$ equals either $C_{I,n}$, or $C_{P,n}$, or $C_{B,n}$ depending on the frame type.

The sum of the number of bits allocated to all frames within a superframe should be equal to the number of bits allocated to that superframe, *i.e.*,

$$T_t = \sum_{n=1}^{N_L} T_{t,n} \tag{11}$$

The technique uses an equal quantizer value Q for all frames in all streams, in order to achieve uniform picture quality. However, taking into account the different picture types (I, P and B), the quantizer values for each frame are related to the fixed Q by

a constant weighting factor:

$$Q_{t,n} = K_{t,n} Q \tag{12}$$

where $K_{t,n}$ is simply either K_I, K_P or K_B, depending only on the frame type.

Combining (10), (11) and (12), results in the following bit allocation equation for frames within a super frame:

$$T_{t,n} = \frac{\frac{1}{K_{t,n}} C_{t,n}}{\sum_{n=1}^{N_L} \frac{1}{K_{t,n}} C_{t,n}} T_t \tag{13}$$

This equation expresses that frames from different streams are allocated bits proportional to their estimated complexities.

To accommodate prioritization of streams as discussed above, the existing techniques may be extended as follows:

One may generalize (12) by including stream priorities p_n as follows:

$$Q_{t,n} = \frac{K_{t,n} Q}{p_n} \tag{14}$$

where p_n are chosen such that:

$$\sum_{n=1}^{N_L} \frac{1}{p_n} = N_L \tag{15}$$

For example, if all streams have the same priority, $p_n = 1$ for all n, such that (15) holds. Higher priority streams are assigned values p_n greater than 1, while lower priority streams are assigned values of p_n smaller than 1.

Combining (10), (11) and (14), one obtains:

$$T_{t,n} = \frac{\frac{p_n}{K_{t,n}} C_{t,n}}{\sum_{n=1}^{N_L} \frac{p_n}{K_{t,n}} C_{t,n}} T_t \tag{16}$$

which can be used for bit allocation instead of (13). From (16), it can be seen that the priorities can be used to allocate more bits to frames from high priority streams and less bits to frames from low priority streams. This strategy implicitly attempts to minimize the distortion criterion (5). Note that this extension applies to both encoding and transcoding.

In the approach described above, intended for encoding, encoding complexities C of frames are estimated from past encoded frames. These estimates are updated every frame and used to allocate bits to upcoming frames. That is, the estimate of complexity for the current frame t and future frames is based on the measurement of the values of the quantizer used in a previous frame as well as the actual amount of bits spent in that previous frame (in the same stream n). Therefore, the estimate is:

$$C_{t,n} = S_{t-\tau,n} Q_{t-\tau,n} \tag{17}$$

where S indicates the number of bits actually spent on a video frame, t indicates the current frame and t-τ indicates the nearest previous frame of the same type (I, P or B), and the prime indicates that the estimate is computed from the output of the encoder. Note again that in reality only 3 different values for these estimates are kept for a single stream, one for each picture type.

While this approach can also be used in transcoding, the present inventor determined that it is possible to improve these estimates. The reason is that in the case of transcoding, one has information about the complexity of the current frame, because one has this frame available in encoded form at the input of the transcoder. However, it has been observed that complexity of the output frame is not the same as the complexity of the input frame of the transcoder because the transcoder changes the rate of the bitstream. It has been observed that the ratio of the output complexity over the input complexity

remains relatively constant over time. Therefore, an estimate of this ratio based on both input and output complexity estimates of a previous frame can be used to scale the given input complexity value of the current frame, to arrive at a better estimate of the output complexity of the current frame:

$$C'_{t,n} = \frac{S'_{t-\tau,n} Q'_{t-\tau,n}}{S_{t-\tau,n} Q_{t-\tau,n}} S_{t,n} Q_{t,n} \qquad (18)$$

where S and Q without prime are computed from the input bitstream.

The approach described above for multi-stream encoding all assumed a constant target bit rate, *i.e.*, a constant bit rate channel. This assumption actually does not hold in certain networks, especially for wireless channels, as previously described. Accordingly, a modified approach that takes into account the time varying nature of the channel is useful.

An extension can be applied if one can assume that the target bit rate varies only slowly relative to the duration of a (super) GOP. This may be the case when the actual channel conditions for some reason change only slowly or relatively infrequently. Alternatively, one may only be able to measure the changing channel conditions with coarse time granularity. In either case, the target bit rate can not be made to vary more quickly than a certain value dictated by the physical limitations. Therefore, the target bit rate can be modeled as a piece-wise constant signal, where changes are allowed only on the boundaries of a (super) GOP.

This approach can be combined with the aforementioned approach by providing a new value of T_s to the bit allocation algorithm (possibly with other extensions as discussed above) at the start of every (super) GOP. In other words, T_s is allowed to vary on a (super) GOP-by-GOP basis.

Another extension is to use the concept of virtual GOPs for the case where the target bit rates varies quickly relative to the duration of a (super) GOP, *i.e.*, the case where adjustments to the target bit rate and bit allocation must be made on a (super) frame-by-frame basis. The use of virtual GOPs was explained for the single-stream dynamic rate

adaptation above. In the multi-stream case, the concept of virtual GOPs extends to the concept of virtual super GOPs.

Another bit allocation approach in joint coding of multiple streams in a LAN environment, such as those based on IEEE 802.11, is suitable for those networks that have multi-rate support. In this case an access point in the gateway may be communicating at different data link rates with different client devices. For this, and other reasons, the maximum data throughput from the gateway to one device may be different from the maximum throughput from the gateway to another device, while transmission to each device contributes to the overall utilization of a single, shared, channel.

As before, there are N_L devices on a network sharing available channel capacity. It may be assumed there are N_L streams being transmitted to these N_L devices (1 stream per device). The system employs a multi-stream manager (*i.e.*, multi-stream transcoder or encoder manager) that is responsible for ensuring the best possibly quality of video transmitted to these devices.

It may be assumed that a mechanism is available to measure the bandwidth or maximum data throughput H_n to each device $n = 1, 2, \ldots, N_L$. In general, this throughput varies per device and varies with time due to variations in the network: $H_n(t)$. It can be assumed that the data maximum throughput can be measured at a sufficiently fine granularity in time. The maximum data throughput H_n is measured in bits per second. Note that the maximum throughput H_n is actually an average over a certain time interval, *e.g.*, over the duration of a video frame or group of frames.

In the case of 802.11 networks, for instance, the bandwidth or maximum data throughput for device n may be estimated from knowledge about the raw data rate used for communication between the access point and device n, the packet length (in bits), and measurements of the packet error rate. Other methods to measure the maximum throughput may also be used.

One particular model of the (shared) channel is such that the gateway communicates with each client device n for a fraction f_n of the time. For example, during a fraction f_1 of the time, the home gateway is transmitting video stream 1 to device 1, and during a fraction f_2 of the time, the gateway is transmitting video stream 2 to device 2, and

so on. Therefore, an *effective* throughput is obtained from the gateway to client n that is equal to:

$$f_n H_n.$$

The following channel constraint holds over any time interval:

$$\sum_{n=1}^{N_L} f_n \leq 1.0 \tag{19}$$

i.e., the sum of channel utilization fractions must be smaller than (or equal to) 1.0. If these fractions add up to 1.0, the channel is utilized to its full capacity.

In the case of transcoding, let R_n denote the rate of the original (source) video stream n. To be able to transmit video streams to all devices concurrently, there may exist a set of f_n, n = 1, 2, ..., N_L, such that the following holds for all n, under the constraint of (19):

$$f_n H_n \geq R_n \tag{20}$$

If such a set of f_n does not exist, then the rate of one or more video sources be reduced. Let R'_n, denote the rate of the transrated (output) video stream n. To retain the highest possible video quality, the minimum amount of rate reduction should be applied, in order for a set of f_n to exist, such that the following holds for all n, under the constraint of (19):

$$f_n H_n = R_n \tag{21}$$

In the case of joint encoding (instead of joint transcoding), the goal is simply to find a solution to (21), under the constraint of (19), where R'_n denotes the rate of the encoder output stream n.

In general, the problem of determining a set of fractions f_n is an under-constrained problem. The above relationships do not provide a unique solution. Naturally, the goal is to find a solution to this problem that maximizes some measure of the overall quality of all video streams combined.

An embodiment is based on a joint coding principle, where the bit rates of different streams are allowed to vary based on their relative coding complexity, in order to achieve a generally uniform picture quality. This approach maximizes the minimum quality of any video stream that are jointly coded, *i.e.*, this approach attempts to minimize distortion criterion (5).

One may consider N_L video streams, where each stream is MPEG-2 encoded with GOPs of equal size N_G. One may also consider a set of N_L GOPs, one from each stream, concurrent in time. This set, also called super GOP, contains $N_L x N_G$ video frames. The first step in some bit allocation techniques is to assign a target number of bits to each GOP in a super GOP, where each GOP belongs to a different stream n. The allocation is performed in proportion to the relative complexity of each GOP in a super GOP. The second step in the bit allocation procedure is to assign a target number of bits to each frame of the GOP of each video stream.

Let T_n denote the target number of bits assigned to the GOP of stream n (within a super GOP). Let $S_{n,t}$ denote the number of bits generated by the encoder/transcoder for frame t of video stream n. The total number of bits generated for stream n over the course of a GOP should be equal (or close) to T_n, *i.e.*,

$$T_n = \sum_{t=1}^{N_G} S_{n,t} \qquad (22)$$

As in the MPEG-2 TM5 a coding complexity measure for a frame is used that is the product of the quantizer value used and the number of bits generated for that frame, *i.e.*,

$$C_{n,t} = Q_{n,t} S_{n,t} \qquad (23)$$

Therefore, (22) can be rewritten as:

$$T_n = \sum_{t=1}^{N_G} \frac{C_{n,t}}{Q_{n,t}} \qquad (24)$$

As in equation (14) a generally constant quality approach may be used. All quantizer values may be equal, up to a constant factor $K_{n,t}$ that accounts for the differences in picture

types (I, P, and B) and a stream priority p_n. Therefore, (24) can be rewritten as:

$$T_n = \frac{p_n}{Q} \sum_{t=1}^{N_G} \frac{C_{n,t}}{K_{n,t}} \tag{25}$$

To achieve (21), the following may hold:

$$T_n = f_n H_n \frac{N_G}{frame\_rate} \tag{26}$$

Combining equations (25) and (26), together with (19), provides the following solution for the set of n unknowns, f_n (factoring out Q).

$$f_n = \frac{\frac{p_n}{H_n} \sum_{t=1}^{N_G} \frac{C_{n,t}}{K_{n,t}}}{\sum_{n=1}^{N_L} \frac{p_n}{H_n} \sum_{t=1}^{N_G} \frac{C_{n,t}}{K_{n,t}}} \tag{27}$$

It is assumed that the channel is utilized to its maximum capacity, *i.e.*, the sum of channel utilization fractions adds up to exactly 1.0. Note that the approach is still valid if the utilization fractions need to add up to a lower value than 1.0. Equation (27) would simply be modified with an additional factor to allow for this. For instance, there may be non-AV streams active in the network that consume some of the channel capacity. In the case of non-AV streams, some capacity has to be set aside for such streams, and the optimization of the rates of AV streams should take this into account, by lowering the mentioned sum lower than 1.0.

Given f_n, the actual target rate for each GOP can be computed with (26).

As mentioned above, the second step in the bit allocation procedure is to assign a target number of bits to each frame of the GOP of each video stream. This can be achieved using existing bit allocation methods, such as the one provided in TM5. Subsequent coding or transcoding can be performed with any standard method, in this case any encoder/transcoder compliant to MPEG-2 (see FIG. 12).

Although the above method has been derived specifically for the wireless LAN case, it should be noted that the above model and equations hold for any other type of LAN or network where a central gateway, server, or access point may communicate with multiple client devices at different maximum rates.

In the case of dynamic rate adaptation, the maximum throughput rates H_n vary in time. In this case, the above method can be combined with the notion of virtual GOPs, or *virtual super GOPs*, which consist of virtual GOPs of multiple AV streams, and overlap in time (see FIG. 13). Equation (27) would be executed at every frame time, to assign a target number of bits to a virtual GOP of a particular stream n. Subsequently, a target number of bits for each frame within each virtual GOP must be assigned, using, for instance, equations (3).

Note further, that the above method can be applied in the case where GOPs are not used, *i.e.*, the above method can be applied on a frame-by-frame basis, instead of on a GOP-by-GOP basis (see FIG. 14). For instance, there may be cases where only P-type pictures are considered, and rate control is applied on a frame-by-frame basis. In this case, there is a need to allocate bits to individual frames from a set of N_L co-occurring frames from different video streams. The above method can still be used to assign a target number of bits to each frame, in accordance to the relative coding complexity of each frame within the set of frames from all streams.

One embodiment uses a single-stream system, as illustrated in FIG. 7. This single-stream system has a single analog AV source. The analog AV source is input to a processing module that contains an AV encoder that produces a digitally compressed bit stream, *e.g.*, an MPEG-2 or MPEG-4 bit stream. The bit rate of this bit stream is dynamically adapted to the conditions of the channel. This AV bit stream is transmitted over the channel. The connection between transmitter and receiver is strictly point-to-point. The receiver contains an AV decoder that decodes the digitally compressed bit stream.

Another embodiment is a single-stream system, as illustrated in FIG. 8. This single-stream system has a single digital AV source, *e.g.* an MPEG-2 or MPEG-4 bit stream. The digital source is input to a processing module that contains an transcoder/transrater that outputs a second digital bit stream. The bit rate of this bit stream

is dynamically adapted to the conditions of the channel. This AV bit stream is transmitted over the channel. The connection between transmitter and receiver is strictly point-to-point. The receiver contains an AV decoder that decodes the digitally compressed bit stream.

Another embodiment is a multi-stream system, as illustrated in FIG. 9. This multi-stream system has multiple AV sources, where some sources may be in analog form, and other sources may be in digital form (*e.g.*, MPEG-2 or MPEG-4 bit streams). These AV sources are input to a processing module that contains zero or more encoders (analog inputs) as well as zero or more transcoders (digital inputs). Each encoder and/or transcoder produces a corresponding output bitstream. The bit rate of these bit streams are dynamically adapted to the conditions of the channel, so as to optimize the overall quality of all streams. The system may also adapt these streams based on information about the capabilities of receiver devices. The system may also adapt streams based on information about the preferences of each user. All encoded/transcoded bit streams are sent to a network access point, which transmits each bit stream to the corresponding receiver. Each receiver contains an AV decoder that decodes the digitally compressed bit stream.

Channel Bandwidth Estimation

The implementation of a system may estimate the bandwidth in some manner. Existing bandwidth estimation models have been primarily based on the estimation of the network capacity over a distributed network of interconnected nodes, such as the Internet. Typically there are many interconnected nodes, each of which may have a different bandwidth capacity. Data packets transmitted through a set of relatively fast nodes may be queued for transmission through a relatively slow node. To attempt to estimate the bottleneck bandwidth over a communication network a series of packets may be transmitted from the server through a bottleneck link to the client. By calculating the temporal spacing between the received packets, the client may estimate the bandwidth of the bottleneck node. Accordingly, the temporal spacing of packets occurs as a result of a relatively slow network connection within the many network connections through which the data packets are transmitted. This temporal spacing does not measure a rate of change of the network bandwidth in terms of a relatively short time frame, such as less than 1 second, but rather is a measure whatever link is the network bottleneck when measured on

a periodic basis, such as every few minutes. Moreover, the physical bottleneck node has a tendency to change over time as the traffic across the distributed network changes, such as the Internet.

Other techniques for estimating the bandwidth of distributed networks involves generating significant amounts of test data specifically for the purpose of estimating the bandwidth of the network. Unfortunately, such test data presents a significant overhead in that it significantly lowers the bandwidth available for other data during the test periods. In many cases the test data is analyzed in an off-line manner, where the estimates are calculated after all the test traffic was sent and received. While the use of such test data may be useful for non-time sensitive network applications it tends to be unsuitable in an environment where temporary interruptions in network bandwidth are undesirable, and where information about link bandwidth is needed substantially continuously and in real time.

It would also be noted that the streaming of audio and video over the Internet is characterized by relatively low bit rates (in the 64 to 512 Kbps range), relatively high packet losses (loss rates up to 10% are typically considered acceptable), and relatively large packet jitter (variations in the arrival time of packets). With such bit rates, a typical measurement of the bandwidth consists of measuring the amount of the packet loss and/or packet jitter at the receiver, and subsequently sending the measured data back to the sender. This technique is premised on a significant percentage of packet loss being acceptable, and it attempts to manage the amount of packet loss, as opposed to attempting to minimize or otherwise eliminate the packet loss. Moreover, such techniques are not necessarily directly applicable to higher bit rate applications, such as streaming high quality video at 6 Mbps for standard definition video or 20 Mbps for high definition video.

The implementation of a system may be done in such a manner that the system is free from additional probing "traffic" from the transmitter to the receiver. In this manner, no additional burden is placed on the network bandwidth by the transmitter. A limited amount of network traffic from the receiver to the transmitter may contain feedback that may be used as a mechanism to monitor the network traffic. In the typical wireless implementation there is transmission, feedback, and retransmission of data at the MAC

layer of the protocol. While the network monitoring for bandwidth utilization may be performed at the MAC layer, one implementation of the system described herein preferably does the network monitoring at the APPLICATION layer. By using the application layer the implementation is less dependent on the particular network implementation and may be used in a broader range of networks. By way of background, many wireless protocol systems include a physical layer, a MAC layer, a transport/network layer, and an application layer.

When considering an optimal solution one should consider (1) what parameters to measure, (2) whether the parameters should be measured at the transmitter or the receiver, and (3) whether to use a model-based approach (have a model of how the system behaves) versus a probe-based approach (try sending more and more data and see when the system breaks down, then try sending less data and return to increasing data until the system breaks down). In a model-based approach a more optimal utilization of the available bandwidth is likely possible because more accurate adjustments of the transmitted streams can be done.

The parameters may be measured at the receiver and then sent back over the channel to the transmitter. While measuring the parameters at the receiver may be implemented without impacting the system excessively, it does increase channel usage and involves a delay between the measurement at the receiver and the arrival of information at the transmitter.

MAC LAYER

Alternatively, the parameters may be measured at the transmitter. The MAC layer of the transmitter has knowledge of what has been sent and when. The transmitter MAC also has knowledge of what has been received and when it was received through the acknowledgments. For example, the system may use the data link rate and/or packet error rate (number of retries) from the MAC layer. The data link rate and/or packet error rate may be obtained directly from the MAC layer, the 802.11 management information base parameters, or otherwise obtained in some manner. For example, FIG. 18 illustrates the re-transmission of lost packets and the fall-back to lower data link rates

between the transmitter and the receiver for a wireless transmission (or communication) system.

In a wireless transmission system the packets carry P payload bits. The time T it takes to transmit a packet with P bits may be computed, given the data link rate, number of retries, and a prior knowledge of MAC and PHY overhead (e.g., duration of contention window length of header, time it takes to send acknowledgment, etc.). Accordingly, the maximum throughput may be calculated as P/T (bits/second).

APPLICATION LAYER

As illustrated in FIG. 19, the packets are submitted to the transmitter, which may require retransmission in some cases. The receiver receives the packets from the transmitter, and at some point thereafter indicates that the packet has been received to the application layer. The receipt of packets may be used to indicate the rate at which they are properly received, or otherwise the trend increasing or decreasing. This information may be used to determine the available bandwidth or maximum throughput. FIG. 20 illustrates an approach based on forming bursts of packets at the transmitter and reading such bursts periodically into the channel as fast a possible and measure the maximum throughput of the system. By repeating the process on a periodic basis the maximum throughput of a particular link may be estimated, while the effective throughput of the data may be lower than the maximum.

Referring to FIG. 21, the technique for the estimation of the available bandwidth may be based upon a single traffic stream being present from the sender to the receiver. In this manner, the sender does not have to contend for medium access with other sending stations. This single traffic stream may, for instance, consist of packets containing audio and video data. As illustrated in FIG. 21, a set of five successful packet transmissions over time in an ideal condition of a network link is shown, where Tx is the transmitter and Rx is the receiver. It is noted that FIG. 21 depicts an abstracted model, where actual transmission may include an acknowledgment being transmitted from the receiver to the transmitter, and intra-frame spacings of data (such as prescribed in the 802.11 Standard). In the actual video data stream having a constant bit rate, the packets are

spaced evenly in time, where the time interval between data packets is constant and determined by the bit rate of the video stream, and by the packet size selected.

Referring to FIG. 22 a sequence of five packets is shown under non-ideal conditions. After the application has transmitted some of the packets, the transmitter retransmits some of the packets because they were not received properly by the receiver, were incorrect, or an acknowledgment was not received by the transmitter. The retransmission of the packets automatically occurs with other protocol layers of the wireless transmission system so that the application layer is unaware of the event. As illustrated in FIG. 22, the first two packets were retransmitted once before being properly received by the receiver. As a result of the need to retransmit the packets, the system may also automatically reverts to a slower data rate where each packet is transmitted using a lower bit rate. The 802.11a specification can operate at data link rates 6, 9, 12, 18, 24, 36, 48, or 54 Mbps and the 802.11b specification can operate at 1, 2, 5.5, or 11 Mbps. In this manner the need for retransmission of the packets is alleviated or otherwise eliminated.

Referring to FIG. 23, the present inventors considered the packet submissions to be transmitted from the application, illustrated by the arrows. As it may be observed, there is one retransmission that is unknown to the application and two changes in the bit rate which is likewise unknown to the application. The application is only aware of the submission times of the packets to be transmitted. The packet arrivals at the application layer of the receiver are illustrated by the arrows. The packets arrive at spaced apart intervals, but the application on the receiver is unaware of any retransmission that may have occurred. However, as it may be observed it is difficult to discern what the maximum effective bandwidth is based upon the transmission and reception of the packets shown in FIG. 23.

After consideration of the difficultly, the present inventor determined that to effectively measure the bandwidth available, a group of packets should be provided to the transmitter in a manner as fast as the transmitter will accept the packets or otherwise without substantial space between the packets in comparison with the normal speed at which they would be provided to the transmitter, for the average bit rate of the video. Referring to FIG. 24, the burst of packets is preferably a plurality, and more preferably 3 or

more. Subsequently, the packets are submitted by the transmitter through the wireless network, to the receiver. At the receiver, the arrival of the individual packets in each burst is timed.

In many cases, the transmission of packets across the wireless network is at an average data rate less than the maximum data rate. Accordingly, the transmitter may buffer a plurality of packets together temporarily that is received for transmission. The data packets may arrive at the transmitter portion of the system at regular intervals, for example, if they come from a video encoder or transcoder that is operating at a constant bit rate. After the buffering, the transmitter may attempt to send the buffered packets as a burst, i.e., at the maximum sending rate. The transmitter may continue to buffer additional groups of packets to form additional bursts, as desired.

One desired result of sending packets in such bursts is that the overall throughput of the system is approximately equal to the target bit rate of the streaming video. The effective throughput, E, can be modified by controlling the following three parameters:

(1) The packet size (e.g., in the number of bytes), or the size of the data payload of each packet; and/or

(2) The number of packets in each burst of packets; and/or

(3) The time interval between subsequent burst of packets.

By way of example, if a payload size of 1472 bytes, and the number of packets in the burst is 10, and the time interval between bursts is 40 milliseconds, the effective throughput is: 10 (packets per burst) x 1472 (bytes per packet) x 8 (bits per byte) / 0.040 (seconds per burst) = 2,944,000 bits per second, or approximately 2.9 Mbps. Therefore, an audiovisual stream with a bit rate of 2.9 Mbps can be streamed at that rate using that wireless connection. It may be observed, that the packets are the actual video signal and not merely additional test traffic imposed on the network. In addition, the system may sequentially transmit the packet bursts in a manner such that the average data rate matches (within 10%) the video bit rate. In this manner, the system may have a continuous measurement of the available video bandwidth, while permitting the average video rate to remain unchanged.

The effective throughput of a system is always lower than (or equal to) the bandwidth or maximum throughput that the channel can support. Bandwidth or the maximum throughput may be denoted as T. For example, it is known that in ideal conditions an 802.11b link in DCF mode can support a maximum throughput (bandwidth) of approximately 6.2 Mbps - if the payload size is 1472 bytes per packet, and the underlying link rate is 11 Mbps. In non-ideal conditions this maximum throughput or bandwidth will be lowered, due to re-transmissions and lowering of link rates. Naturally, the effective throughput can never exceed the maximum throughput. The ideal situation is illustrated in FIG. 24 while the non-ideal situation is illustrated in FIG. 25. In the case shown in FIG. 24 the channel actually may support a higher througput, up to a maximum throughput, of $T=T_A$ Mbps. Therefore, $E_S < T_A$ and there is space for additional traffic. In the case shown in FIG. 25 the maximum throughput drops because the underlying MAC uses more of the capacity of the channel to transmit the packets in the data stream and there is less room for additional traffic. The maximum throughput in this case, say $T=T_B$ is lower than the case shown in FIG 24: $T_B < T_A$. The effective throughput can still be supported: $E_S < T_B$ holds as well.

It is the maximum throughput or bandwidth T that is estimated, in order to provide the transmitter with the right information to adapt the audio/video stream bandwidth (if necessary). The maximum throughput is achieved, albeit temporarily, during transmission of a packet burst. Therefore, the maximum throughput is estimated by computing the ratio of the number of bits transmitted during a burst, and the time duration of that burst. More precisely, consider a burst of N packets ($N \geq 2$) arriving at the receiver, where packet i , $1 \leq i \leq N$, in that burst arrives at time t_i (in seconds). Note that at the receiver it may not know the time at which the sender submitted the first packet to the network for transmission. As shown in FIGS. 24 and 25, during the time interval $\Delta t = t_N - t_1$ between the arrival of the first and the last packet of a burst, the network is busy transmitting packets 2 to N (all packets in the burst except the first). If one assumes that each packet in a burst carries the same payload P bits, then the amount of bits transmitted during the interval Δt is equal to P * (N-1) bits. Therefore, the maximum throughput or bandwidth at the time of this burst is:

$$T = \frac{P \bullet (N-1)}{t_N - t_1}$$

More generally, one may denote the maximum throughput or bandwidth for a burst j by T_j. The payload of packets during burst j is P_j (all packets during a burst have the same payload). The number of packets in burst j is N_j. The time of arrival of packet i in burst j is $t_{j,i}$ and the time interval measured for burst j is $\Delta t_j = t_{j,N} - t_{j,1}$.

It is noted that the receiver may also utilize time stamps measured at the sender side, if the sender embeds those time stamps into the packet payload. The packet payload may also include a packet sequence number. The packet payload may also include a burst sequence number. Such sequence numbers may be used by the receiver to detect packet losses. The packet payload may also contain a field that indicates that a packet is the first packet in a burst, and/or a field that indicates that a packet is the last packet in a burst, and/or a field that indicates that a packet is neither the first nor the last packet in a burst.

Timestamps or measurements of time and time intervals can be provided by clocks internal to the hardware/software platform. Note that different hardware/software platforms may offer different APIs and may support clocks with different performance in terms of clock resolution (or granularity). For example, on a Linux platform, the gettimeofday() API is available, which provides time values with microsecond resolution. As another example, on a Windows 2000 / Windows XP platform (Win32 API), the GetTickCount() and QueryPerformanceCounter() APIs are available. The latter API can be used to retrieve time values with sub-millisecond resolution. The actual resolution of the time values provided by the QueryPerformanceCounter() API depends on the hardware. For example, this resolution was found to be better than microsecond resolution on two different Windows 2000 laptop PCs, and was found to be better than nanosecond resolution on a newer Windows XP desktop PC.

The bandwidth measurements may be done on an ongoing basis, that is, more than just once. Every burst of data packets during the streaming of audio/video data

may be used to estimate bandwidth available during transmission of that burst. Such measurements performed at the receiver are sent back to the sender.

A test setup was implemented using software running on two Windows 2000 laptop PCs, both equipped with IEEE 802.11b WLAN client cards. These WLAN cards on these laptops were configured to communicate in the 802.11 ad-hoc mode, and the IP protocol settings were configured to create a 2 laptop private network. Software running on one PC acted as a server, sending packets over the network to the receiver using the UDP, IP and 802.11b protocols. Note that UDP may be used instead of TCP, as UDP is more suitable for real-time traffic. It is noted that the system may use other protocols, such as for example, the Powerline Communication networks or other LANs.

The first example illustrates throughput performance of 802.11b in a generally ideal case, where the laptop PCs were located close to each other, and interference from external sources was minimized. The 802.11b cards were configured to communicate at the maximum 11 Mbps link rate. The packet payload was constant at 1472 bytes (an additional 28 bytes are used by UDP and IP, such that the 802.11 MAC delivered 1500 byte packets). Each experiment consisted of transmission of 100 bursts. In this example, each burst consisted of 10 packets and the time between subsequent bursts was scheduled to be 40ms. Therefore, effective throughput in this case is approximately 2.9 Mbps.

Results for the ideal conditions are shown in FIG. 26. From other measurements, it is known that the maximum throughput/bandwidth in this case is 6.2 Mbps on average. Note that the bandwidth varies somewhat around the 6.2 Mbps value; the average value over 100 bursts is 6.24 Mbps and the standard deviation is 0.22 Mbps. The average value over 100 burst is very close to the expected value, and the standard deviation is reasonably small. Methods to handle the variations are discussed in the next section.

The second example illustrates throughput performance of 802.11b in generally non-ideal conditions, where the laptop PCs were located much further away from each other, at a distance of 43 m, in an indoor environment containing many cubicles, and

a few walls between the sender and receiver. All other parameters were the same as in the first example.

Results for the non-ideal case are shown in FIG. 27. The maximum throughput in this case is much lower: 3.3 Mbps on average over 100 bursts. The standard deviation over 100 bursts is much higher: 1.20 Mbps. The diagram shows the decreased throughput performance and the increased variations in performance (note that the vertical axis has a different scale in FIGS. 26 and FIG. 27).

From this second example, it is noted that the variation in measured bandwidth values may be a useful parameter in itself to use as feedback in an audio/video streaming system — as an alternative to estimating bandwidth directly.

Robustness

Measurements of bandwidth are subject to temporal variations under most conditions, as illustrated in FIGS. 26 and 27. Some of these variations, generally referred to as noise, are not meaningful in the context of wireless broadcasting. It was determined that one source of errors is caused by the (limited) resolution of the clock used to measure packet arrival times. With such errors present it is desirable to provide a robust estimate of (instantaneous) bandwidth that can be used as input to a rate adaptation mechanism at the audio/video encoder at the sender.

There exists a trade-off between the number of packets in a burst and the robustness of the bandwidth estimate. Robustness of the bandwidth estimate can be increased by using a larger number of packets per burst. For example, using bursts with more than two packets reduces the effects of limited resolution of the measurement clock. However, increasing the number of packets per burst means the buffer size at the sender side must be increased, resulting in higher cost of implementation, and higher transmission delays.

The examples shown in FIGS. 26 and 27 assumed a burst size of 10 packets; however, any suitable number of packets may be used. Some temporal variations in the estimates of bandwidth typically remains as the number of packets is increased to its practical maximum. The processing of either the bandwidth estimates or of the measured

time intervals may be used to reduce the variations. Processing techniques may be applied to compute a final estimate of the bandwidth .

Traditional techniques applicable to measuring Internet bottleneck bandwidth use a frequency distribution (histogram) of a set of bandwidth estimates and take either the mean, median or mode(s) of that distribution as the final bandwidth estimate. The approach is partly based upon viewing the data as a set of data to be collected, and thereafter processed. However, the present inventor has determined that such techniques are not appropriate for real-time bandwidth estimation in WLANs. One of the principal reasons the present inventor determined that such techniques are inappropriate is that it takes several seconds before enough bandwidth samples can be collected to form a meaningful frequency distribution. However, in the case of video over a wireless network, the channel is subject to variations on a much smaller (subsecond) timescale and the system should be able to respond to those changes in a much faster manner.

To overcome this limitation of a set-based premise, the present inventor determined that the data should be analyzed as a sequential set of measurement samples, as opposed to viewing the data as a set of data to be collected, and thereafter processed. In this manner, the temporal nature of the data becomes important. The data may be treated as a set of measurement samples as a time sequence, i.e., as a discrete time signal. Accordingly, if the samples are received in a different order the resulting output is different. Assuming the measurement samples are spaced equidistantly in time, various signal processing techniques can be applied to eliminate "noisy" variations, including but not limited to the following.

(1) FIR filtering. Non-recursive filtering with a finite number of filter tabs. One example is a moving average filter. FIGS. 26 and 27 illustrate the effect of a 10 tab moving average filter on the sequence of bandwidth measurement samples.

(2) IIR filtering. Recursive filtering with a finite number of filter tabs. One example is a first-order recursive filter that weights both the previous

estimate with the current measurement sample to compute a new estimate. FIGS. 26 and 27 illustrate the effect of a first order IIR filter on the sequence of bandwidth measurement samples.

(3) Statistical processing. Mean square error (MSE) estimates, maximum a posteriori (MAP) estimates, Wiener filtering, Kalman filtering. Statistical processing provides a particularly convenient framework, because it allows one to both filter samples and predict future values as the same time. Forming a prediction is important since the results of the measurements are used to control the rate of audio/video data transmitted in the (near) future.

(4) Curve fitting. Fitting curves, such as straight lines, splines, and other, allows interpolation, approximation and extrapolation from the noisy data samples. Curve fitting is especially useful in case the measurement samples are not spaced exactly equidistantly in time.

In each of these methods, the additional processing to compute a final estimate for the bandwidth at burst j may utilize a limited number of bandwidth samples from the past T_m, $j-M_1 \le m \le j$, and may also utilize a limited number of final bandwidth estimates from the past T_m^*, $j-M_2 \le m \le j-1$. One embodiment may, for example, utilize a first order IIR type of filter as follows:

$$T_j^* = (1-w) \bullet T_{j-1}^* + w \bullet T_j$$

where w is a parameter between 0 and 1. For instance, if w = 0.5, the final estimate of bandwidth at burst j is computed by weighting equally the previous final estimate of the bandwidth at burst j-1 and the current bandwidth sample for burst j. The parameter w controls the amount of smoothing or averaging applied to the bandwidth samples, where the amount of smoothing is high when w is low, and the amount of smoothing is low when w is high. This parameter may be held constant; alternatively, one may change this

parameter adaptively. This technique was used in the examples in FIGS. 26 and 27, where the value of w was 0.1.

Note that instead of filtering or processing bandwidth samples T_j, one may also filter or process measured time intervals Δt_j before computing bandwidth values using $T = \frac{P \bullet (N-1)}{t_N - t_1}$. In that case, one may utilize samples of measured time intervals intervals from the past Δt_m, $j-M_1 \leq m \leq j$, as well as a limited number of processed time interval estimates from the past Δt_m^*, $j-M_2 \leq m \leq j-1$, to compute a final estimate of a representative time interval for burst j, Δt_j^*. Then, one may apply $T = \frac{P \bullet (N-1)}{t_N - t_1}$ using this final estimate of the time interval, to compute a final estimate of the bandwidth at burst j, T_j^*. One example is to use IIR filtering on the measured time intervals:

$$\Delta t_j^* = (1-w)\, \Delta t_{j-1}^* + w\, \Delta t_j$$

followed by:

$$T_j^* = \frac{P_j \bullet (N_j - 1)}{\Delta t_j^*}$$

Such filtering, estimation and prediction techniques allow filtering out an appropriate amount of noisy variations, while still providing a final estimate quickly.

The measurement results at the receiver are transmitted back to the sender. The sender uses this feedback information to adapt the audio/video being transmitted, especially its rate. The feedback information transmitted from receiver to sender may consist of an estimate of (instantaneous) bandwidth/maximum throughput as computed by the receiver. It may also include raw measurement results, or a partial estimate, which the sender may use to compute a final estimate. It may also include a time stamp, indicating the time at which the estimate was computed, and a sequence number, indicating the packet

number or packet burst number. The feedback information may also contain an indication of whether the receiver detected any packet loss.

Feedback information can be sent back to the sender using the same network link that carries the audio/video data. In particular, packets with feedback information can be transmitted between transmission of packet bursts with audio/video data from sender to receiver. Such feedback packets may be sent periodically, for example, after every burst of audio/video packets, or after every K bursts, or whenever desired, for example, only if there is a significant change of the bandwidth performance. The feedback packets only uses a small portion of the available bandwidth for transmission. This overhead should be minimized, i.e., kept small while still allowing the sender to react in a timely fashion. The amount of information in such feedback packets is very small compared to the audio/video data, therefore the bandwidth overhead is very small. Still, the sender may take this small overhead into account in its bandwidth allocation and adaptation strategy.

Referring to FIG. 28A, a flow diagram for an exemplary receiver is shown. The receiver receives the packet bursts j and determines packet losses and measures the arrival of the packets. Then the receiver computes the bandwidth sample for the burst j. Thereafter, the receiver may compute the final bandwidth estimate for the burst j by incorporating the bandwidth of the previous packets. After estimating a final bandwidth the receiver transmits the bandwidth information back to the sender. It is to be understood that the sender may calculate the bandwidth information and bandwidth estimates based upon information from the receiver.

Referring to FIG. 28B, a flow diagram for another exemplary receiver is shown. The receiver receives the packet bursts j and determines packet losses and measures the arrival of the packets. Then the receiver computes the time interval for burst j. Thereafter, the receiver may compute the final bandwidth estimate for the burst j by incorporating the time intervals of the previous packets. After estimating a final bandwidth the receiver transmits the bandwidth information back to the sender. It is to be understood that the sender may calculate the bandwidth information and bandwidth estimates based upon information from the receiver.

Referring to FIG. 29, a flow diagram for an exemplary transmitter is shown. The transmitter sends a packet burst to the receiver. The transmitter then waits a pre-determined time interval to receive feedback from the receiver. When the feedback information is received the transmitter may adapt the rate and schedule the packets to the receiver. It is to be understood that the sender may calculate the bandwidth information and bandwidth estimates based upon information from the receiver.

All references cited herein are incorporated by reference.

The terms and expressions which have been employed in the foregoing specification are used therein as terms of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding equivalents of the features shown and described or portions thereof, it being recognized that the scope of the invention is defined and limited only by the claims which follow.